CLAIMS

Now, therefore, the following is claimed:

1. A computer system for enabling selective execution of sets of code in computer programs, comprising:

memory for storing a computer program, said computer program having a set of code and a branch instruction; and

processing circuitry configured to receive run time data indicating whether said set of code is enabled and to set a value of a mode indicator based on said run time data, said processing circuitry configured to receive and execute said branch instruction during a run of said computer program, said processing circuitry configured to branch to a first address of said memory, in response to said branch instruction and based on said value of said mode indicator, when said set of code is enabled, said processing circuitry further configured to branch to a second address of said memory, in response to said branch instruction and based on said value of said mode indicator, when said set of code is disabled.

2. The system of claim 1, wherein said set of code, when executed by said processing circuitry, tests for errors that occur during said run of said program.

3. The system of claim 1, wherein said branch instruction includes a first address identifier and a second address identifier, said first address identifier identifying said first address and said second address identifier identifying said second address, and wherein said processing circuitry further configured to select one of said address identifiers based on said value of said mode indicator in response to said branch instruction.

4. The system of claim 1, wherein said processing circuitry is further configured to maintain said value of said mode indicator during said run of said computer program and until termination of said run.

5. The system of claim 1, wherein said processing circuitry executes said set of code when said processing circuitry branches to said first address, and wherein said processing circuitry executes an instruction at said second address upon completing execution of said set of code.

6. The system of claim 1, wherein said branch instruction includes an address identifier that identifies said first address, and wherein an instruction of said set of code is stored in said memory at said first address.

7. The system of claim 6, wherein said processing circuitry, in executing said branch instruction, is configured to identify said second address based on said first address when said set of code is disabled.

8. The system of claim 6, wherein said processing circuitry, in executing said branch instruction, is configured to flip a bit of said first address when said set of code is disabled.

9. The system of claim 1, wherein said branch instruction includes an address identifier that identifies said second address, and wherein an instruction of said set of code is stored in said memory at said first address.

10. The system of claim 9, wherein said processing circuitry, in executing said branch instruction, is configured to identify said first address based on said second address when said set of code is enabled.

11. The system of claim 9, wherein said processing circuitry, in executing said branch instruction, is configured to flip a bit of said second address when said set of code is enabled.

12. The system of claim 1, wherein said branch instruction includes an address identifier identifying one of said addresses.

13. The system of claim 12, wherein said branch instruction is encoded with an expression, and wherein said processing circuitry is further configured to compute a new address for each branch instruction that is encoded with said expression and that is executed by said processing circuitry during said run of said computer program, said new address based on an address identifier included in said each branch instruction.

14. A system for enabling selective execution of sets of code in computer programs, comprising:

means storing a computer program, said computer program having a set of code and a branch instruction, said branch instruction including an address identifier identifying a first memory address;

means for receiving, during a run of said program, run time data indicating whether said set of code is enabled;

means for setting a value of a mode indicator based on said run time data;

means for identifying a second memory address in response to said branch instruction; and

means for branching to said second address based on said value of said mode indicator and said identified memory address, said branching means including a means for executing an instruction at said second address.

15. A method for enabling selective execution of sets of code in computer programs, comprising the steps of:

storing a computer program in memory, said computer program having a set of code and a branch instruction, said branch instruction including an address identifier identifying a first address in said memory;

receiving, during a run of said program, run time data indicating whether said set of code is enabled;

setting a value of a mode indicator based on said run time data;

identifying a second address in said memory and in response to said branch instruction;

branching to said second address based on said value of said identifying step and said value of said mode indicator; and

executing an instruction at said second address in response to said branching step.

16. The method of claim 15, wherein said identifying step includes the step of flipping a bit of a bit value representing said first address.

17. The method of claim 15, further comprising the step of maintaining said value of said mode indicator during said run of said computer program and until termination of said run.

18. The method of claim 15, wherein said branch instruction includes an address identifier identifying said second address, said identifying step including the step of selecting said second address identifier based on said value of said mode indicator.

19. The method of claim 15, wherein said branch instruction is encoded with an expression, and wherein said method further includes the step of:

computing a new address for each branch instruction from said computer program that is encoded with said expression and that is executed during said run of said program, said new address based on an address identifier included in said each branch instruction.

20. The method of claim 15, wherein said instruction executed in said executing step is outside of said set of code, and wherein said branching step prevents execution of said set of code, said value of said mode indicator indicating that said set of instructions is disabled.

21. The method of claim 15, wherein said instruction executed in said executing step is included in said set of code, said value of said mode indicator indicating that said set of instructions is enabled.

22. The method of claim 21, further comprising the step of testing, in response to said set of code, for errors generated by said program run.